

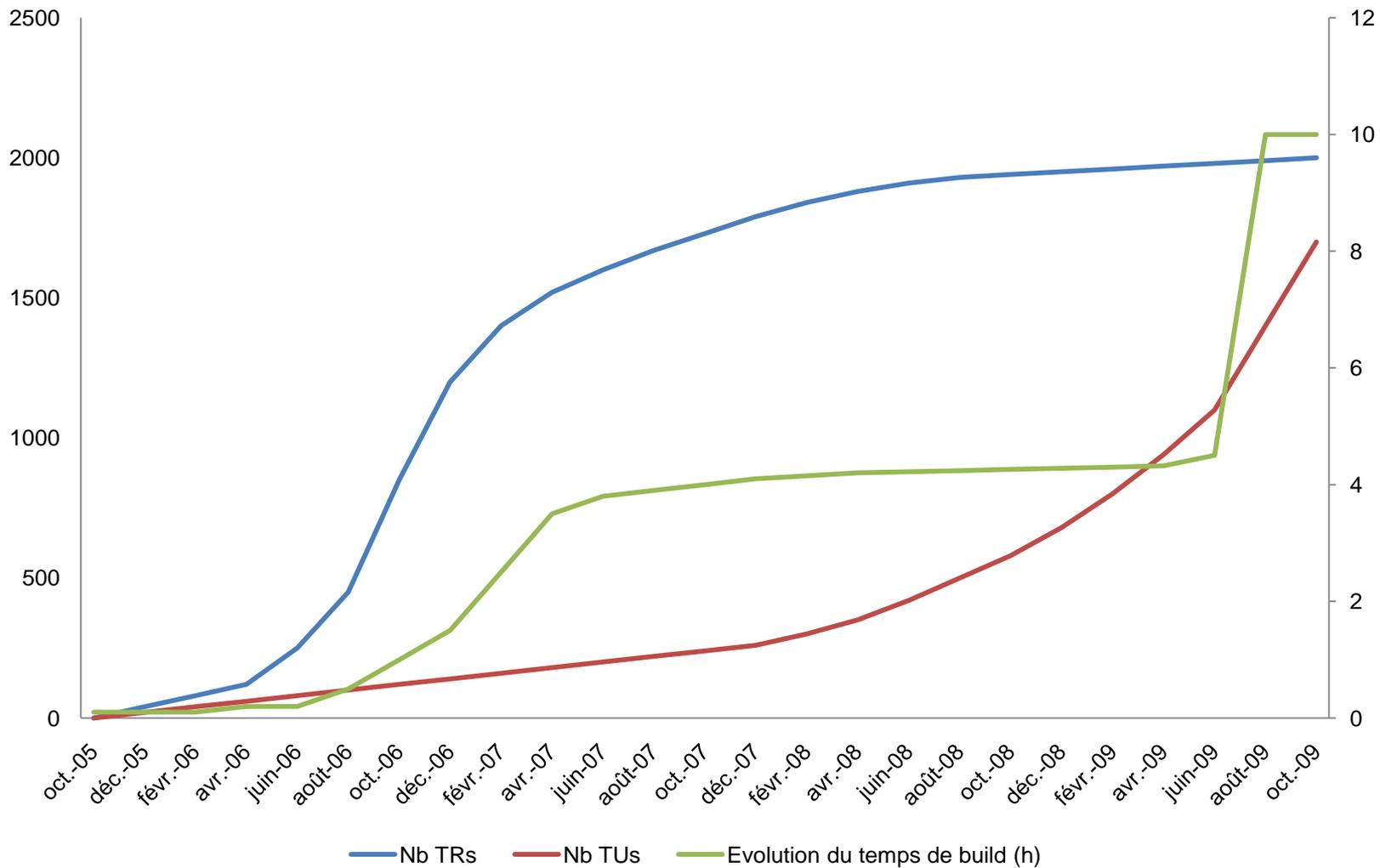
Que tester dans une application ?



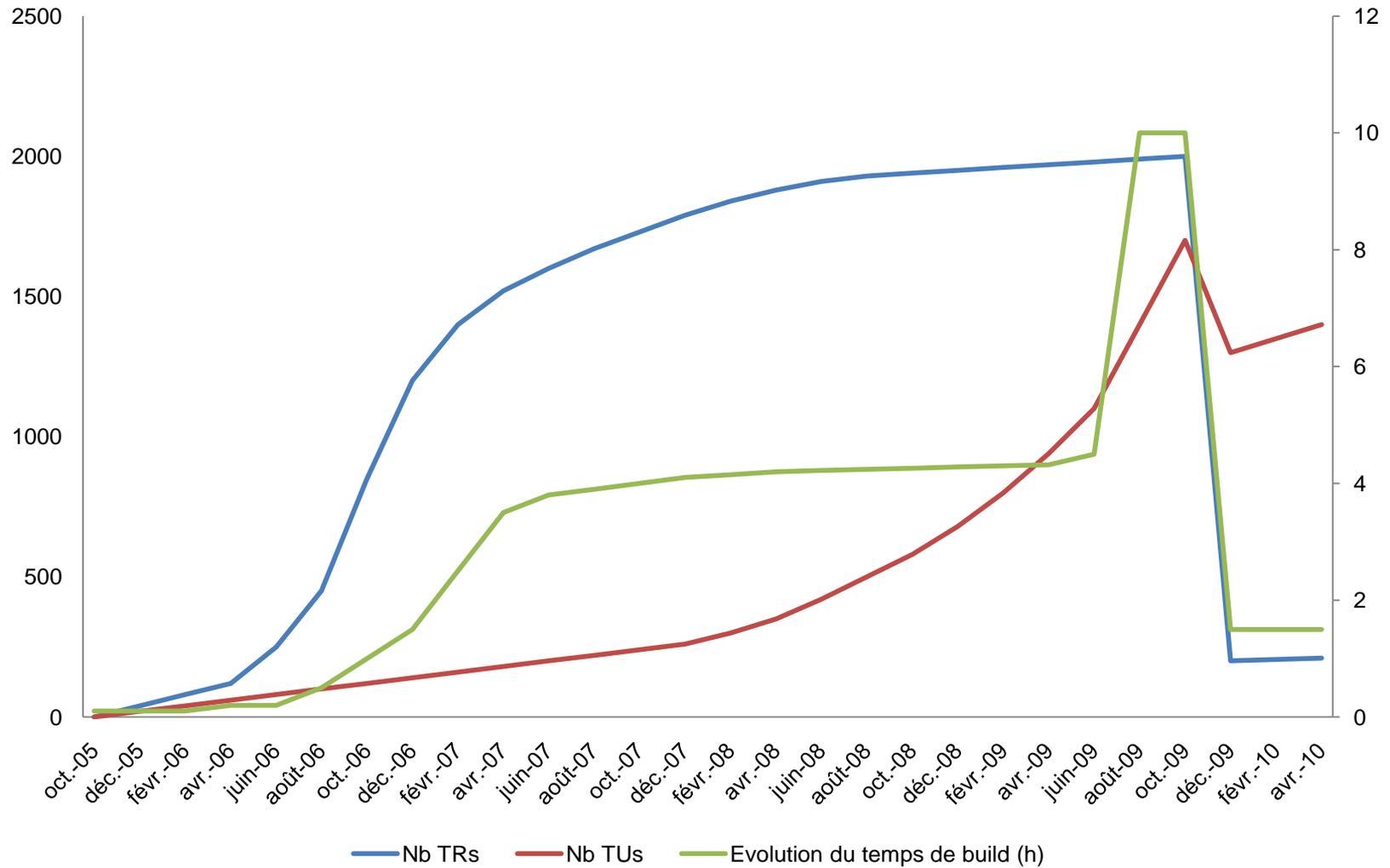
Cette session a été faite sous la forme d'Open Space Discussion, c'est-à-dire une discussion libre où tout participant est également un acteur.

Le contenu de la session a donc été créé par les participants eux-même.

Ce document ne contient donc que quelques exemples de code qui ont été montrés pendant la session.



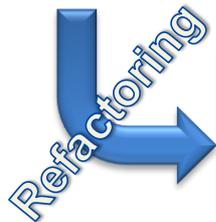
Evolution TRs – TUs – Prévision 2010





```
[TestMethod]
public void DataOriented_CodeDeTestPerduDansBruit()
{
    try
    {
        //Créé un emprunteur gagnant 15.000€ par an, et un prêt de 132.000€ pendant 30 ans
        DataHelper.InsertEmprunteur("Nom", "Prénom", 15000);
        DataHelper.InsertPretHypothécaire("Nom", 132000, 30);

        // Ici : code de test
    }
    finally
    {
        // Attention, faire la suppression dans le sens inverse !
        DataHelper.DeletePretHypothécaires();
        DataHelper.DeleteEmprunteur();
    }
}
```



```
[TestMethod]
[InsertEmprunteur("Nom", "Prénom", RevenuBrutAnnuel = 15000, Priorité = 1)]
[InsertPretHypothécaire(Emprunteur = "Nom", Montant = 132000, DureeAnnee = 30, Priorité = 2)]
public void DataOriented_Declaratif()
{
    // Ici : code de test
}
```

```
[TestMethod]
public void ExempleSimulateurWeb()
{
    using ( SimulateurWeb sw = new SimulateurWeb() )
    {
        sw.VaSurPage("~/Accueil.aspx");

        //1. Créer un emprunteur
        sw.Boutons.ParTexte("Créer un emprunteur");
        sw.Champs.ParId("txtNom").Remplit("Dautrepe");
        sw.Champs.ParId("txtPrenom").Remplit("Pierre-Emmanuel");
        sw.Champs.ParId("txtRevenu").Remplit("15000");
        sw.Boutons.ParTexte("Soumettre");

        //2. Vérifier que la création s'est bien passée
        sw.ValidePage("~/VisualiserEmprunteur.aspx");
        sw.Labels.ParTexte("Nom").Valide("Dautrepe");
        sw.Labels.ParTexte("Prénom").Valide("Pierre-Emmanuel");
        sw.Labels.ParTexte("Revenu Annuel Brut").Valide("15.000,00 €");

        //3. Visualiser la liste des emprunteurs
        sw.Boutons.ParTexte("Voir Liste");
        sw.Tableaux.ParId("gvEmprunteurs")
            .ValideDimension(1, 2)
            .ValideEnTeteColonne("Nom", "Prénom")
            .ValideLigne(0, "Dautrepe", "Pierre-Emmanuel");
    }
}
```

Framework « maison »
au dessus de WATIN

Se passer d'IIS :
Monte un répertoire local sous
forme de « Site Web » dans un
Web Server « maison » light

```
[TestClass]
[StartWebServer("SiteWeb")]
public class ExempleWebServer
{
    [TestMethod]
    public void ExempleSimulateurWeb()
    {
        //Idem Code précédent
    }
}
```

Exemple Test Avec Mock (RhinoMock)



```
[TestMethod]
public void MockInterface()
{
    MockRepository mocks = new MockRepository();

    //1. Création de l'action
    TypeAction action = EntiteFactory.Create<TypeAction>();
    action.RolesAutorises = new[] { "Role2" };

    //2. Création de l'utilisateur et de ses attentes : nous gérons le cas 0, 1, N pour les rôles associés
    IIdentite utilisateur = mocks.StrictMock<IIdentite>();
    Expect.Call(utilisateur.Roles).Return(AucunRoles);
    Expect.Call(utilisateur.Roles).Return(UnRole_Role1);
    Expect.Call(utilisateur.Roles).Return(UnRole_Role2);
    Expect.Call(utilisateur.Roles).Return(PlusieursRoles);

    mocks.ReplayAll();

    //2. Vérification de la méthode
    Assert.IsFalse(action.GetSiActionAutorisee(utilisateur), "Cas où l'utilisateur n'a pas de rôle");
    Assert.IsFalse(action.GetSiActionAutorisee(utilisateur), "Cas où l'utilisateur n'a qu'un seul rôle - Role1");
    Assert.IsTrue(action.GetSiActionAutorisee(utilisateur), "Cas où l'utilisateur n'a qu'un seul rôle - Role2");
    Assert.IsTrue(action.GetSiActionAutorisee(utilisateur), "Cas où l'utilisateur a plusieurs rôles");

    mocks.VerifyAll();
}
```

Utilisation RhinoMock

Spécification des
« Attentes » (réponse
souhaitée du système)

Vérification que les
« Attentes » ont bien été
appelées

Exemple Test avec Mock (Injection de dépendance)



Pouvoir injecter le comportement souhaité pour le test

```
[TestMethod]
public void MockDAL()
{
    /* On va mocker notre DAL, mais cependant garder
    * l'implémentation réelle pour la gestion
    * des utilisateurs
    */
    using ( new ModeMockPour<DAL>() )
    {
        DAL.UseMockPour<IPretDAL>(new MockPretDAL());
        DAL.UseImplementationReellePour(typeof(IUserDAL));

        //1. Récupération des prêts filtrés pour un utilisateur
        var controleur = new ControleurGestionPret();
        var critere = new Critere { Utilisateur = "PDA",
                                   DateDebut = new DateTime(2009, 01, 01),
                                   DateFin = new DateTime(2009, 08, 01) };
        var prets = controleur.GetPretPourCritereRecherche(critere);

        //2. Vérification des données
        Assert.AreEqual(2, prets.Length, "Le filtre n'a pas renvoyé le bon nombre d'éléments");
        Assert.AreEqual(120000, prets[0].Montant, "Le premier prêt renvoyé n'est pas correct");
    }
}

public class MockPretDAL : IPretDAL
{
    public Pret[] GetPretDansPeriode(DateTime dateDebut, DateTime dateFin)
    {
        return new[] { new Pret { Emprunteur = "Plus de 100.000", Montant = 120000, Duree = 20 },
                       new Pret { Emprunteur = "Moins de 60.000", Montant = 50000, Duree = 15 },
                       new Pret { Emprunteur = "Moins de 5 ans", Montant = 3000, Duree = 3 } };
    }
}
```

```
public class ControleurGestionPret
{
    public Pret[] GetPretPourCritereRecherche(Critere critere)
    {
        //1. Récupération des prêts pour le range de date spécifié
        Pret[] prets = DAL.Get<IPretDAL>()
            .GetPretDansPeriode(critere.DateDebut,
                               critere.DateFin);

        //2. Récupération des droits de l'utilisateur
        Droit[] droits = DAL.Get<IUserDAL>()
            .GetDroits(critere.Utilisateur);

        //3. Application du filtre
        Filtreur filtreur = new Filtreur(droits);
        return filtreur.Applique();
    }
}
```

Travailler avec des interfaces pour avoir de l'injection de dépendances



■ Norman

- norman@dotnethub.be
- <http://letsgo-agile.blogspot.com/>



■ Pierre-Emmanuel

- pierre@dotnethub.be
- <http://www.pedautreppe.com>
- <http://pedautreppe.developpez.com>

